LA-5525-M, Vol. 7

September 1982

PIM-7

COMPUTING DIVISION
PROGRAMMER'S INFORMATION MANUAL

c. 1

CTSS Overview

Los Alamos National Laboratory is operated by the University of California for the United States Department of Energy under contract W-7405-ENG-36

Los Alamos
Los Alamos National Laboratory
Los Alamos, New Mexico 87545

CTSS Overview

Produced by the Computer Documentation Group
Additional copies: the Computing Information Center 505-667-6992
Direct questions to the
Computing Division Consulting Office 505-667-5745

PURPOSE

This document is an overview of the major computing capabilities of the Cray Time Sharing System (CTSS) available on the Cray-1 computers at Los Alamos. It also contains a brief description of how CTSS fits into the Los Alamos Computing Network and a discussion of the features that distinguish CTSS from the other computing systems in the Network.

This document does not contain comprehensive descriptions of CTSS capabilities. Rather, it is a guide that points to other documents that will give you more information about a particular capability that is described. This overview does not contain instructions or examples for using CTSS. For this kind of information, you must refer to the CTSS Primer.

CONTENTS

1 INTERFACING WITH C DIVISION

USING THIS CHAPTER

Before you begin reading about CTSS, you might like to read this chapter quickly to find out about organizations in the Computing Division that can assist you when you have questions about any phase of computing at Los Alamos. We strongly recommend that you read the section titled Obtaining More Information: Documentation because it gives you a quick summary of what kinds of documentation are available. If the topics are already familiar to you, please proceed to Chapter 2.

BECOMING A VALIDATED USER

To use CTSS or any of the other systems in the Los Alamos Computing Network, you need a validated user number, which is usually the number on the front of your badge (called a Z number) with a leading zero, and a password. To receive a password and to validate your Z number, you must fill out an ICN User Validation Form that is available from the Computer and Telecommunications Security Office, OS-4 (7-4844), or from the Dispatcher's Desk in the Central Computing Facility (CCF) located on the main floor of building SM-132. Your password will be sent to you in a packet of materials from OS-4; you must return your password receipt before you become validated.

If you intend to do classified computing, which is probable on CTSS, you must obtain a classified password. Classified passwords are also assigned by the Computer and Telecommunications Security Office. To obtain a classified password, you must mark on the ICN User Validation Form that you wish to use the Secure partition and also indicate the highest classification level that you need to use. (See Chapter 2 for descriptions of the Secure partition and classification levels.) Your division leader must sign the ICN User Validation Form if you request to do classified computing. You will be asked to go to OS-4 (Building SM-43, Room A346) to pick up your classified password when it has been assigned. You can possess both a classified and an unclassified password.

OBTAINING HELP: CONSULTING OFFICE

The Computing Division Consulting Office should be your first contact point if you have a question about computing. The Consulting Office, whose phone number is 7-5745, is open Monday through Friday, 8 a.m. to 12 noon and 1 p.m. to 5 p.m., excluding Monday from 1 p.m. to 2 p.m. The Office is located in Room 284 in Building SM-200.

Members of the Consulting Office can help you choose the systems and related software that will best meet your computing needs. They will also assist you in correcting the errors in your programs. They can help you convert programs that you are moving from one computing system to another.

Many times a question that you have can be answered over the phone. A quick call can determine if a trip to the Consulting Office is necessary. If you do go to the Office, be sure to take all the listings, input, and output related to your computing problem.

OBTAINING MORE INFORMATION: DOCUMENTATION

Because some CTSS topics are covered only briefly in or are omitted from this Overview, you will have to consult a number of documents. The CTSS Catalog of Documentation (Catalog CTSS-99) is your best reference for determining which documents contain the information you need. The Catalog contains a list of the available CTSS user documents and lists them according to computing functions. By knowing what kind of computing activity you need to do, you can then find the appropriate document. There are also two monthly publications that describe the most recent changes and improvements in CTSS and the other Los Alamos computing systems: the ICN Change Bulletin and the Computing Division News.

All computer documentation, except for the Computing Division News, is available in two forms: hardcopy and online. You can pick up hardcopy documentation from the Computing Information Center (CIC), which is located in Room 257 in Building SM-200. You can also call the CIC, whose phone number is 7-6992, and the librarian will place you on the appropriate mailing lists and also send you any documentation you need.

The online documentation is immediately available to you when you have signed on to CTSS. There are two categories of online documentation: temporary and permanent. You must access these two categories differently.

Temporary online documentation, which is announced in the ICN Change Bulletin, is made available when changes occur and permanent documentation cannot be immediately updated, or when the need to disseminate information is time-critical. This documentation can be accessed using the INFO utility (INFO CTSS-3 writeup).

Permanent documentation is stored on the Common File System (CFS). (See Chapter 2 to see how the CFS fits into the computing capabilities that are available at Los Alamos.) You can access permanent documentation by executing the MASS utility, which will make local copies of the documents on the CTSS machine you are using. You must then convert the files to a form called CTSS native text by executing the NTEXT utility (CTSS-7 writeup). Once the conversion is done, you can scan the file using one of the editors mentioned in Chapter 3 of this overview.

The specific steps you need to follow to access either INFO or the permanent documentation on CFS are shown on the first page of the CTSS Catalog (Catalog CTSS-99) and on the first page of the ICN Change Bulletin.

LEARNING MORE: EDUCATION PROGRAM

The Computing Division Education Program offers both classroom and self-instructional courses. The available courses are advertised in the Computing Education Program Catalogue and the Computing Division News. If you have any questions about the course offerings, call the Computing Education Program Coordinator at 7-3193.

Courses are usually determined by user needs and interests. They are designed to help you compute more efficiently and also to keep you up-to-date with changes and improvements made in the computing systems.

2 BASIC CTSS CONCEPTS

USING THIS CHAPTER

Chapter 2 is designed to give you an idea of how CTSS fits into the Los Alamos Integrated Computing Network (ICN) and also to introduce some CTSS concepts. We recommend that you read this entire chapter if you are a new user. If you are somewhat familiar with CTSS, you may want to proceed to subsequent chapters.

WHAT IS CTSS?

The Cray Time Sharing System (CTSS) is the interactive operating system that has been installed on the Cray Research Inc., Cray-1 computers at Los Alamos. Currently, there are 4 Cray-1s, locally named by letters: Machines V, W, X, and Y. A fifth Cray-1 is expected to arrive in the fall of 1982 and will be named Machine Z.

HOW CTSS FITS INTO THE NETWORK

The Big Picture

As we mentioned above, the computing network at Los Alamos is called the Integrated Computing Network (ICN). Throughout this document, we refer to the ICN as the Network.

The physical location of the heart of the Network is called the Central Computing Facility, or CCF. Many Network components, or nodes, reside outside the CCF. These are called remote computers and range in complexity from single computers used for obtaining output, submitting programs, or or entering information to subnetworks of computers called distributed processors. The remote computers that can be used for obtaining output are called CBTs and are described briefly in Chapter 4.

The Cray-1 machines on which CTSS is running are located in the CCF. These machines are used primarily to run users' programs and are called worker machines or simply workers. There are several other types of worker machines: Control Data Corporation 7600s and Cyber series machines. Operating

systems other than CTSS are used on these machines, so we are not concerned with them in this document. Although, the system that runs on the 7600s, called the Livermore Time Sharing System, is very similar to CTSS; we mention the similarities in the systems in later sections of this chapter.

CTSS is an interactive timesharing operating system. You communicate with the CTSS machines by entering information at your terminal. This information is intercepted by several layers of computers, called Keyboard Communications Concentrators (KCCs) and Synchronous Communications Concentrators (SYNCs), to name a few, before it reaches the worker. Most of the communications between your terminal and these intermediate levels of computers are transparent to you. However, you may receive a message from a KCC when you attempt to sign on to a CTSS worker. These messages are described in the Interface document (Interface CTSS-1).

The Network contains separate nodes for other computing activities such as generating output, storing information, and controlling the execution of programs in a production environment. These activities are handled by the Print and Graphics Express Station (PAGES), the Common File System (CFS), and the FOCUS production control system, respectively. The programs that you can execute to output a file on the PAGES devices are described briefly in Chapter 4. The MASS utility allows you to send files to and get files from the CFS and is mentioned in the Files section of this chapter. The PROD utility used to submit production programs to the FOCUS production system, which runs noninteractive programs during the day, evenings, and weekends, is described in Chapter 5.

There are many other nodes in the Network. Figure 1 shows the complete layout of the Network. You will find the Cray-1s in the left-central portion of the drawing.

A utility called INFORM can be used to report the status of several of the computers in the Network. With the INFORM utility, you can query to find out the following things about any CTSS machine: whether the machine is up or down, the memory and CPU usage, the number of terminals signed on to the machine, and how many production jobs are currently active. You can also query to find out the status of the CFS. INFORM is described in the INFORM CTSS-24 writeup.

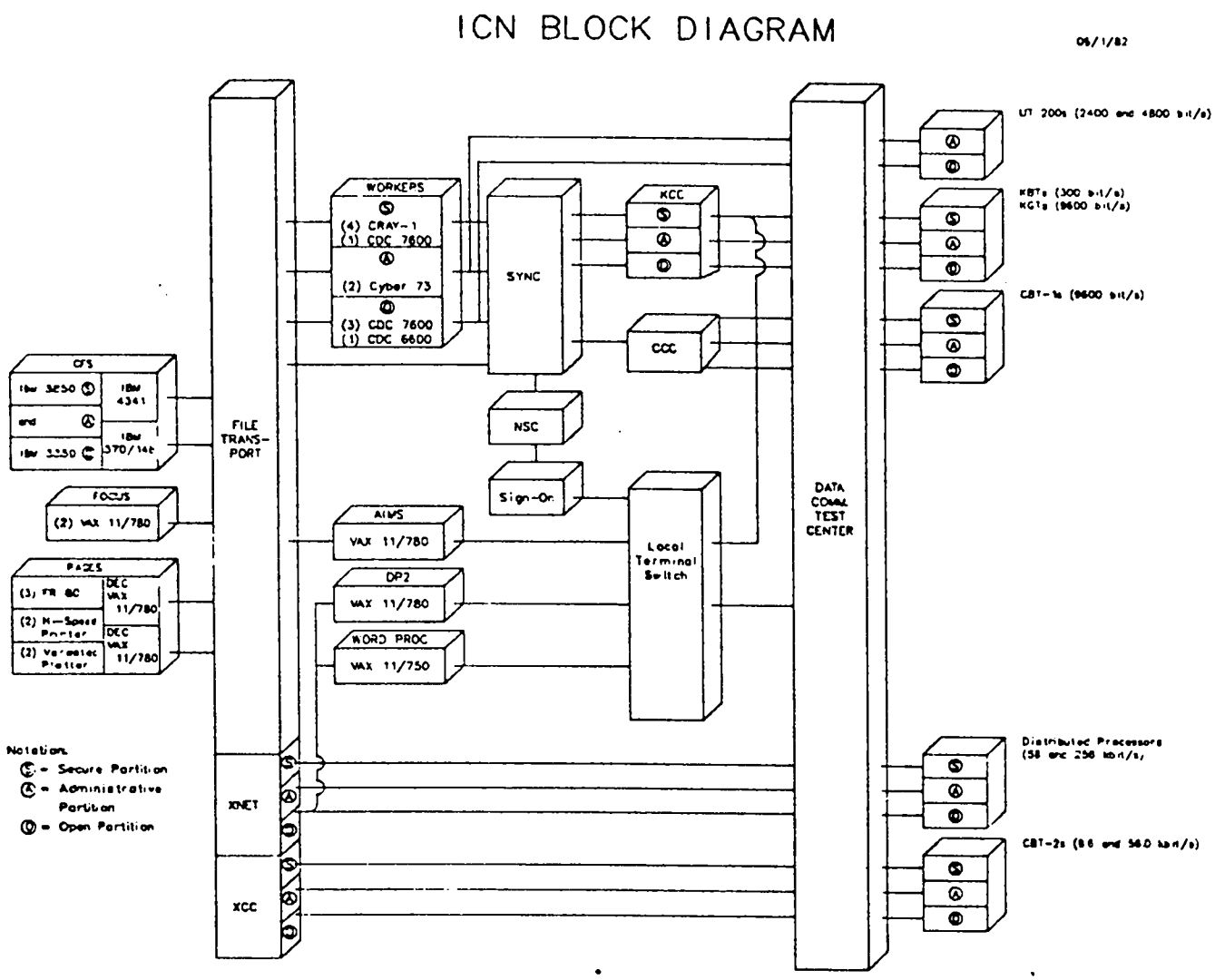

Figure 1 Los Alamos Computing Network

A Partitioned Network

The Network processes a wide variety of data, including sensitive unclassified and classified data. For this reason, the Network is divided into three separate computing partitions that safeguard and maintain the integrity of the data being computed. The names of the partitions are Secure (S), Administrative (A), and Open (O). The Secure partition is the least accessible partition while the Open partition is the most readily accessible partition.

Currently, all the Cray machines are part of the Secure computing partition. Within the Secure partition, classified, as well as unclassified, data can be processed. You must have a Q-clearance and the approval of your division leader to use CTSS for classified computing. You must also have a classified password.

Now, if you take a minute and look back at Figure 1, you will see that all components within the Network reside within a particular partition or partitions. Notice that the terminal nodes (at the far right of the drawing), the KCCs, and the CFS have all been marked with the letters S, A, and O showing the partitions, just as the workers have been marked.

Classification Levels Within the Partitions

The allowed classification levels of computing vary with each partition. Unclassified computing is permitted in all partitions, and is the only type of computing allowed in the Open partition. In the Administrative partition, primarily sensitive information is processed. The Secure partition is the only partition in which the following levels of computing are permitted.

- Secret
- Confidential
- Protect As Restricted Data (PARD)

We recommend that you check with your group leader, the person in your group who is responsible for computing, or Group OS-4 (7-4884) to learn about the restrictions involved when using particular classification levels.

CTSS FEATURES — IN BRIEF

This section describes some of the fundamental features of CTSS that distinguish it as an operating system. Many of these attributes are not available on any other Network operating system. Most, however, are also attributes of the Livermore Time Sharing System (LTSS) that is available on the CDC 7600 machines at Los Alamos.

Before we begin with these specific features, we would like to mention some general Cray-1 features.

Key Features

Perhaps the two primary features of the Cray-1, as compared to the other workers in the Network, are its large memory resources and its automatic vectorization capabilities. These features provide definite advantages.

The amount of memory resources on the Cray-1s is machine-dependent and ranges from 0.8 to 3.8 million words. The memory section of the Cray-1 CPU is not divided into large core and small core. These two attributes may allow you to write programs without overlays.

The Cray-1 is called a vector machine because it efficiently processes sets of values rather than many values one-at-a-time. Most of this type of processing is performed automatically by the CFT compiler without your knowledge. However, you can benefit more from the vectorization by choosing efficient algorithms and using good programming techniques. We can give you two suggestions about using DO loops: use short loops containing no IF, CALL, or GO TO statements and watch how you nest loops because only the innermost DO loops are vectorized.

If you want specific information about how to take advantage of both the memory resources and the vectorization, please check with the Consulting Office (7-5745). The consultants can make some suggestions about how you can use these important features.

Interactive Terminal Access

CTSS provides both interactive and batch processing. The interactive features of the system allow for editing and debugging activities to be done most efficiently from a terminal. You can sit at your terminal, execute your programs, and then query and alter the programs as they are executing. Batch computing allows for the running of large programs that perform several hours of computations,

sometimes called number crunching, primarily during the evenings and on weekends.

If you are going to be running codes interactively, you should acquaint yourself with the dynamic debugging utility called DDT. This is a powerful tool that allows you to monitor programs as they are running or to perform an analysis after a program has aborted. See the DDT document (DDT CTSS-124) for a complete description of the utility's capabilities.

Signing On

A standard signon line is in use for all worker systems. You can read about the signon sequence in either the CTSS Primer or the Interface document (Interface CTSS-1).

If you are going to be doing classified computing on CTSS, however, you should be aware of a few things. Make sure that you have received your classified password from Group OS-4 (7-4844) and are familiar with the password. You will have to sign on at a properly protected terminal, a terminal connected to the ICN by protected wirelines or encryption devices. When you are doing classified computing, you should always be aware of the classification level at which you are computing. Special commands called control key entries can be used to check on your classification level. (Control key entries are described in a following section.)

Signing Off

The control key entry <CTRL-D> allows you to sign off of CTSS. (Control key entries are described later in this chapter.) See the Interface document (Interface CTSS-1) for specific information about CTSS responses when you sign off.

Suffixes

Once you have signed on to CTSS, you can perform five computing tasks simultaneously through five separate command streams. Each of these streams is called a suffix. An example of using suffixes is that while compiling, loading, and executing a program on one suffix, you can edit a different program on another suffix.

The suffixes have letter names: A, B, C, D, and E. After signon you are automatically connected to suffix A. One of

the control key entries described in the Interface CTSS-1 writeup can be used to switch between suffixes.

Files

There are two types of files on CTSS: local and public. Local files are generally accessible to only the user who created them. You create a local file when you compile and execute a program or use an editor, to name two examples. Local files are not permanent. Local files with write access are destroyed after 8 hours if they are not used, while local files without write access are destroyed after 17 hours if they are not used. (File access, such as read and write access, is described in the Files and Character Set document -- CTSS-2.)

Public files are generally provided with CTSS and are maintained by Computing Division personnel or people outside of the division. They are available to a large number of users; some are accessible to all CTSS users. These files have indefinite lifetimes.

All aspects of CTSS files, including their format and attributes, are described in the Files and Character Sets document (CTSS-2, also called Files and Character Sets). Formats of files are particularly important if and when you ship files between the CTSS machines and different workers in the Network. Files to be shipped between computers of different types should be in either standard text or standard print format. The utilities that you can use to control the format of files are SPRINT (SPRINT CTSS-8 document), STEXT (STEXT CTSS-9 document), and NTEXT (STEXT CTSS-7 document). To ship files between computers, use the MOVE (MOVE CTSS-119 writeup) or MAGIC (MAGIC CTSS-91 writeup) utility.

CTSS contains techniques for organizing files. The LIB utility combines several files into a single large file, thus saving file space and making it easier to maintain the files. LIB is described in the LIB CTSS-118 writeup. The main system library on CTSS called CFTLIB (described in the following chapter) generates families of files. Families of files must be merged into one file by using a utility called COMPACT before they can be input to CTSS utilities. (COMPACT is described in the COMPACT CTSS-111 writeup.)

There is a built-in feature that allows you to give copies of all, or a subset, of your local files to another user on your CTSS machine. See the GIVE utility writeup (GIVE CTSS-116) to learn about this feature.

Local and public files do not satisfy the need for permanent file storage. The Common File System (CFS) is the Network's centralized permanent file storage system. It consists of an IBM 3850 mass storage system and two IBM controllers. It is your responsibility to send local files that you wish to keep to the CFS; otherwise, you are in danger of losing information. Execute the MASS utility to ship files to and from the CFS. You can store your files in hierarchical repositories called directories, making them as complex as you choose. See the CFS Primer and the Programmer's Information Manual (PIM) Volume 5 to learn how to access the CFS.

Controller-Controllee Relationships

Two buzz words that you will read frequently in CTSS and LTSS documentation are controller and controllee. A controller is generally anything that controls or runs a program; you, sitting at your terminal, or a computer program can function as a controller. A controller runs a controllee. Typically, a controllee is an executable code.

On CTSS, you can establish many controller-controllee relationships and thereby run programs without having to be directly involved. In a typical controller-controllee chain, a program is both the controller of the program beneath it and the controllee of the program above it. This situation is common when running large programs in the CTSS production environment. All kinds of activities can be happening simultaneously: one controller interacts with FOCUS; a main program begins executing; this executing program calls up the MASS utility to retrieve a file from CFS; MASS retrieves the file; and so on.

In a controller-controllee chain, many levels of controllers and controllees exist at one time but only one controllee can be in execution and in memory at one time. You can set up communication pathways through which the controllers and controllees communicate. This ability to establish computing processes gives CTSS a great deal of flexibility.

Drop Files

Drop files are a special type of local file and are essentially "restart" files. If CTSS crashes or your program reaches a time limit that you have set, you can use the drop file of the program you were running to continue the program from the point at which you were interrupted.

CTSS writes the drop file each time your program is rolled out of memory. This file contains the executing image of

your program just at the time that it was dumped out of memory.

Control Key Entries

Control key entries are special sequences of keys that enable you to query CTSS for information and to send information to CTSS. They can also be used to send information to the Keyboard Communications Concentrators (KCCs).

Most often, control key entries are entered by depressing the CTRL key while simultaneously depressing another key that enters a lowercase character. In CTSS and LTSS documentation, these keys are displayed in a particular notation, for example, <CTRL-E>. This notation means that you type the lowercase letter E while depressing the CTRL key. You do not type the angle brackets (<>); they serve only as punctuation to show that a control key entry is a composite.

The Interface document (Interface CTSS-1) contains a complete list of control key entries. We recommend that you become familiar with and use the control key entries. They are most useful when you are uncertain about the conditions under which you are interacting with CTSS or when you want to change a particular system condition.

Execute Line

When you execute programs on CTSS or LTSS, whether they are your own or system-supplied, you must use a standard input format called an execute line, which requests that the system execute some file. Please note that in general, CTSS expects input in lowercase letters. The form of the execute line is as follows.

```
filename parameters / t p
or
filename parameters
```

where

filename — is the name of the local or public file containing the program to be executed.

parameters — is any input information that may be required to execute the program.

t specifies the maximum amount of CTSS time required to run the program.

p specifies the priority at which you wish to run the program.

The t and p portions of the execute line are important because their values affect how your bank account will be charged for your CTSS usage. See the following section for more information about CTSS resources. For more specific information about the CTSS execute line, such as the default and allowable values for t and p, refer to the Interface writeup (Interface CTSS-1).

Bank Accounts

CTSS resources are allocated in repositories called bank accounts that contain CTSS priority-weighted time. The Director's Office of the Laboratory, your division office, and the Computing Division have jointly agreed on your account's allocation of time. This account of time will generally be shared by a single group or sometimes an entire division.

CTSS priority-weighted time, measured in bank points, is the product of the machine time used (not the time limit specified in the execute line) and the priority specified. For example, if machine time used equals 1.5 minutes and you specified a priority of 2 in the execute line, 3 bank points are subtracted from your account.

Use CTSS time carefully. If you specify high priorities, your bank account will be depleted quickly. If your bank account runs out of time, all programs running under that account are automatically reset to the minimum priority. If you type an execute line when your bank account is zero, your requested priority will automatically be reduced to the minimum.

Bank accounts are reset at the beginning of each bank period. Bank periods and their starting times are as follows.

BANKDAY	0800 hours each workday
BANKNITE	1730 hours each worknight except Friday
BANKWKND	1730 hours Friday

HOLIDAY 1730 hours on the workday preceding a holiday

We urge you to discuss your bank account allocation with the person in your group who is most knowledgeable about computing. We also recommend becoming familiar with a utility called SNOOPY (CTSS-122) that you can run to monitor your bank account usage. By carelessly specifying high priorities, you may deplete your bank account and prevent your colleagues from completing their computing tasks.

3 BASIC TOOLS

USING THIS CHAPTER

Chapter 3 presents brief descriptions of the editors, compiler, assembler, loader, and various libraries that are available on CTSS to create, load, and run your computer programs. At the end of the chapter, several general-purpose controllers that can be used to oversee the running of your programs are described briefly.

For the most part, the descriptions give the basic capabilities of these tools. In some cases, the advantages and disadvantages of the tools will be included.

EDITORS

Text editors are interactive programs for creating and modifying text, which includes primarily source codes and data files. The two major CTSS editors are FRED and ED, which offer you two different approaches to editing text.

FRED provides a vast reservoir of editing capabilities with over 300 commands. Each command performs a separate function. In addition to standard line editing, FRED contains special capabilities for file manipulation, register setting, data loading, column referencing, IF testing, DO looping, and execution of procedure-type files. FRED executes quickly and thus works well with very large files.

The prospect of learning over 300 commands is undoubtedly awesome; however, you can learn a subset of the total commands and do your editing adeptly. FRED is documented in the FRED CTSS-17 writeup; there is also a FRED Preprimer that will give you the basics of using FRED.

ED was designed to be concise and yet powerful. It consists of a small core of commands, fewer than twenty in all, that can be used together in a complementary fashion. Thus, ED contains a manageable number of commands to learn. Included in these commands is a pattern searching capability that is unmatched by any other capability available with other editors. You can also execute an existing file containing ED commands and text.

There are two disadvantages of using ED. First, ED is not designed to handle columns of information readily. Second,

it takes a long time to open a large file. This situation will be improved as the library that ED uses is optimized. You will probably not notice the speed problem if you are editing relatively small files.

ED is part of the Software Tools collection of programs and libraries that were implemented by Lawrence Berkeley Laboratory (LBL) from an original set of software developed by Kernighan and Plauger. There are a number of ED references: ED Primer (ST-6 writeup) and ED (ST-7 writeup). An ED Tutorial that was written by one of the creators of ED is also available in the Software Tools package of documents. UNIX documentation on ED may also be useful.

COMPILER

The CFT (Cray Fortran) compiler is the primary CTSS compiler. It translates Fortran language statements into Cray-1 machine code. CFT accepts as input an extended version of Fortran called Cray-1 Fortran. It performs the automatic vectorizaton of inner DO loops, as we mentioned in Chapter 2.

To invoke the CFT compiler, you must enter an execute line that consists of a CFT control statement. Options are available to specify the names of the source file, relocatable binary file, and output listing file. Options are also available to turn on the production of a cross-reference map and a symbol table. If you intend to debug a program, you can turn off the code optimizer and the vectorizing features that would have been used by default.

The CFT compiler and the Cray-1 Fortran language are described in the following documents.

- Fortran (CFT) Reference Manual (SR-0009) published by Cray Research, and
- CFT Reference Card published by Cray Research.

A controller called RCFT allows you to compile, load, and if you choose, execute a program all in one step. It automatically invokes the CFT compiler and LDR loader if you request both compiling and loading operations. For more information about RCFT, see the RCFT CTSS-23 document.

ASSEMBLER

The assembly language for CTSS is called CAL (Cray Assembly Language). Using CAL, you can symbolically express all of the hardware functions of the Cray-1 that are available to the user. The language statements include symbolic machine instructions, a set of powerful pseudo instructions, and instructions for defining your own macros. Consider using CAL only if you need to make specific use of the Cray-1 hardware; using a high-level language such as Fortran is considerably easier than learning CAL.

Once you have written a CAL program, you can input your CAL modules to the CAL assembler through the use of a control statement. After the assembler passes through your program making all the needed conversions to machine functions, you can then execute the LDR loader to load the program into memory. Finally, you can execute the loaded version.

To learn more about CAL and the assembler, see the following two documents.

- Cray-1 CAL Assembler Reference Manual (SR-0000) published by Cray, and
- CAL Reference Card published by Cray.

LOADER

LDR is the CTSS loader for loading relocatable binary files that have been produced by the CFT compiler or CAL assembler. LDR will also accept special libraries that you have built using the BUILD utility (described in a later section in this chapter). LDR produces an executable controllee, a symbol table, and a load map. The symbol table can be used if you need to do some debugging with the DDT utility (described in the DDT CTSS-124 writeup).

LDR also has features for force loading, squeezing out common blocks, ordering of common blocks, and presetting core. All of these features plus naming the relocatable binary file, needed libraries, and controllee are handled in the LDR execute line. You have many options for determining the nature of the load map. Rather than include all of this information in the LDR execute line, you can create a directive file containing all of the instructions to LDR.

LDR will load both segmented and nonsegmented programs. On some computing systems, these programs are referred to as overlaid and nonoverlaid codes. If you have a large file and are thinking about dividing it into segments by setting

up overlays before you compile and load the program, you do not need to do so on CTSS. Instead, can keep your source file intact and include directions for the segmentation, called TREE directives, in your LDR directive file. LDR will then automatically load your program in segments.

To learn more about LDR, see the LDR CTSS-117 writeup.

LIBRARIES

There are a number of CTSS subroutine libraries that perform general-purpose computing functions, mathematical operations, file input and output, and graphics functions. In the descriptions that follow, we have divided the libraries into categories. NOTE: the graphics libraries are discussed in Chapter 4.

Some of the libraries are default libraries and will be searched automatically. If you choose to use the optional libraries, you must specify them in the LDR execute line. You also have the option to build your own libraries.

General-Purpose Libraries

Name: CFTLIB

Status: Default library

Contents: System library for the CFT compiler. Routines needed to perform I/O and other operations called for in Fortran statements such as PRINT, REWIND, and BACKSPACE; service routines except for graphics and mathematical routines.

Reference: Writeup CFTLIB W001 and two files called MANUAL and FTNMAN on CFS under the /USERPOOL pathname. Call the Consulting Office (7-5745) to find out how to scan or obtain hardcopies of these two files.

Mathematical Libraries

There are three main mathematical libraries that are public files: CFTMATH, CALMATH, and CRAYMATH. There is also a mathematical and statistical library called IMSL (see the IMSL description).

Name: CFTMATH

Status: Default library

Contents: Primarily mathematical routines that have been written in Fortran and are also usable on other computing systems.

Reference: Guide to the Program Library and Abstracts (PIM-2)

Name: CRAYMATH

Status: Default library

Contents: Elementary mathematical functions for use with the CFT compiler. Originally written by Cray Research but adapted for use on CTSS.

Reference: Cray-1 Mathematical Subroutines Reference Manual (document number 2240014).

Name: CALMATH

Status: Optional library

Contents: Specialized high-performance mathematical routines.

Reference: Guide to the Program Library and Abstracts (PIM-2)

Name: IMSL

Status: Optional; available from CFS in three forms; stored under three separate CFS pathnames:
/PROGLIB/CTSS/IMSLIB (binary version)
/PROGLIB/CTSS/IMSLSRC (source version)
/PROGLIB/CTSS/IMSLPL (source library created by the HISTORIAN program)

Contents: Proprietary mathematical and statistical library with subroutines written in Fortran. Property of IMSL, Inc.

Reference: IMSL Reference Manuals

Building Your Own Libraries

Using the BUILD utility, you can create your own libraries that can then be loaded by LDR. Consider learning about and using BUILD if your programs perform a set of operations or tasks regularly that could be programmed easily as a library of subroutines. You can input the compiled versions of your routines to BUILD. The libraries that you create can be set up so that they use other CTSS libraries such as CFTLIB or CFTMATH. See the CTSS-104 writeup for more information about BUILD.

CONTROLLERS

In Chapter 2, we defined a controller as a program that runs another program, or as a user at a terminal who executes programs. Several general-purpose controllers on CTSS for managing and creating procedure files are listed and described briefly below. Note that they vary greatly in complexity and in what kinds of functions they perform.

- COSMOS. COSMOS is used primarily to run production jobs in a noninteractive environment by allowing you to set up job control files that are made up of COSMOS commands. Consequently, COSMOS is a language as well as a utility that formats and uses the language.

  Job control files can be set up to run other programs, or other controllers and system utilities. There are also some built-in arithmetic and character manipulation functions in COSMOS that may be used in the job control file.

  COSMOS can be run in an interactive mode from a terminal, which allows you to take advantage of several CTSS features: control key entries and CTSS utilities such as MASS and the output utilities mentioned in the next chapter.

  For more information about COSMOS, see the COSMOS CTSS-108 document. COSMOS is also mentioned in Chapter 5.

- XEQ. To use XEQ, you input a procedure-type file containing any number of controllee execute lines. The controllees are executed in succession. You can pass messages to the running controllee.

  Within XEQ, you can run any CTSS utility you choose. There are some built-in features for allowing a controllee to bypass XEQ and send messages directly to your terminal.

For more information about XEQ, see the XEQ LTSS-542 document. (XEQ usage is identical on CTSS and LTSS.)

- CTL. CTL contains a superset of COSMOS capabilities. CTL operates in both batch and interactive modes. It is a very powerful interactive controller. You can also execute many CTSS utilities inside CTL almost as though you were in the main operating system environment. You even have access to several arithmetic functions and text substitution features. For more information about CTL, see the CTL document (unnumbered).

4 GETTING INFORMATION IN AND OUT

USING THIS CHAPTER

This chapter focuses on the major types of information that are processed by computers: text and graphics. Brief descriptions of software used to produce these types of information are included. We also discuss some specific considerations about generating classified information. We briefly tell you where you will find your textual and graphics output that has been generated by the Print and Graphics Express Station (PAGES) devices. At the end of the chapter, we mention magnetic-tape usage.

TEXTUAL INFORMATION

In this manual, textual information refers to source files, data files, and listings. Using one of the editors mentioned in Chapter 3, you can generate source and data files that can then be input to a compiler or an assembler. During the stages of compiling, loading, and executing, you will obtain a number of different kinds of listing files that you can use to monitor the progress of your program's execution.

Types of Output and Device Options

For text output, you have two choices for the form of your output: hardcopy or microfiche. Hardcopies can be obtained at your terminal, from a printer attached to a computer called a Computer Based Terminal (CBT) that is located in your technical area, or from the laser printers that are part of the PAGES equipment located in the CCF. Microfiche output is produced through PAGES on film recorders.

File Format Considerations

The CTSS text editors create source and data files in a format called native text. The listings that are generated by the CFT compiler, CAL assembler, LDR loader, and text files produced by most programs are also in native text format. You can use the native text format if you will be looking at a file at your terminal. If you are going to be sending a file to a CBT printer, the PAGES printers, or the PAGES film recorders, your file must be converted to standard print format.

The same utilities that send your files to the PAGES output device of your choice will also perform the conversion from native text to standard print format. This saves you from having to do the conversion. The output utilities also contain options for carriage control functions, changing the case of a file, setting the orientation of a page, setting the number of lines per page, and other functions.

Output Utilities

The list that follows gives the name of a particular output device and then a brief description of the utility that you can use to send text files to the device. The list of the utilities is not comprehensive; we have listed those utilities that are the easiest to use.

- Laser printer. HSP converts native text files to standard print files and sends the print files to PAGES for output on a laser printer; see the HSP CTSS-21 document.

- CBT printer. CBT converts native text files to standard print files and sends the print files to the printer of a CBT; see the CBT CTSS-20 document.

- Film recorder. FICHE converts native text files to standard print files and sends the print files to PAGES for fiche output on a film recorder; see the FICHE CTSS-5 document.

- Your terminal. TPRINT formats a native text file for printing at a hardcopy device that is connected to your terminal; see the TPRINT CTSS-27 document. (Use an editor to display a native text file at your terminal.)

GRAPHICAL INFORMATION

Before we begin our description of graphical information, we wish to encourage you to read the Los Alamos Integrated Graphics System User's Guide (PIM-8). This document is designed to help you choose from a variety of graphics devices and graphics software. It also contains an introductory description of the Los Alamos Graphics System.

Graphical information refers to a pictorial image that can contain text. You create graphics images by executing a graphics utility or by including subroutine calls to a graphics library in your application program. You have the choice of producing graphics images on a graphics terminal

(such as a Tektronix terminal, a VT640 terminal, or an AED terminal); on a Versatec plotter connected to a CBT; or on the electrostatic plotters and film recorders that are part of PAGES. As you will recall, the PAGES film recorders can also be used to generate strictly textual information.

Unless you choose to immediately display graphics images at a graphics terminal, you must create a Common Graphics System (CGS) Metafile. A CGS Metafile is a disk file that stores a device-independent representation of the graphics images. The Metafile can then be translated for output on any of the graphics devices by using a special utility called a CGS Metafile postprocessor.

The sections that follow summarize the graphics software according to three major groups: utilities, libraries, and postprocessors.

Graphics Utilities

There are three graphics utilities available on CTSS: MAPPER, MOVIE, and GAS.

- MAPPER is used to create a wide variety of publication-quality visuals, including text slides, graphs, schematics, charts, and tables. It is described in the GR897 document.
- MOVIE displays the results of three-dimensional finite-element analysis. It is described in the GR834 document.
- The Graphical Analysis System (GAS) was originally designed for use with hydrodynamic programs. It is used to display and analyze the files generated by programs that use a calculational mesh. GAS is described in the GR833 document.

Graphics Libraries

Brief summaries of the graphics libraries follow. None of the graphics libraries are default libraries; you must specify them in your LDR execute line if you choose to use them.

In your programs, along with calling graphics library subroutines, you must also call CGS Device Control subroutines. These subroutines are common to all of the graphics libraries and are used to access and control the

graphics devices and to generate CGS Metafiles. The documentation for CGS Device Control (GR800) is included with the documentation for each graphics library.

Name : CGS Library -- called CGSCFT on CTSS

Status: Optional library

Contents: An integrated set of general-purpose, device-independent graphics primitives. Used as the foundation for developing graphics utilities and higher level graphics subroutines.

Reference: CGS Graphics (GR801)

Name: DISSPLA Library -- called DISCFT on CTSS

Status: Optional library

Contents: Vendor-supplied library used to produce publication-quality graphics. Offers the widest variety of all the libraries of character fonts, alphabets, and line types.

Reference: DISSPLA (GR802) and DISSPLA Pocket Manual

Name: NCAR Library -- called NCARCFT on CTSS

Status: Optional library

Contents: High-level graphics subroutines. Quality rivals DISSPLA but not as flexible or comprehensive as DISSPLA.

Reference: NCAR (GR803)

Name: SC-4020 Library -- Called SCCFT on CTSS

Status: Optional library

Contents: Subroutines that provide emulation of subroutines originally written for the obsolete SC-4020 film recorder. Maintained to ensure compatibility with existing application programs.

Reference: SC-4020 (GR804)

Graphics Postprocessors

Remember that the postprocessors accept only CGS Metafiles as input. The graphics utilities and libraries can generate Metafiles. The list that follows gives the name of the graphics device followed by a description of the software used to output information on that device.

- AED terminal. PCAED translates a Metafile sequentially for display on an AED color graphics terminal. PCAED is described in the GR860 writeup.

- CBT Versatec plotter. PCBT sends a Metafile to a specified CBT for output on a Versatec plotter. PCBT is described in the GR861 writeup.

- PAGES electrostatic plotter. PESP sends a Metafile to PAGES for output on an electrostatic plotter. PESP is described in the GR862 writeup.

- PAGES film recorder. PFILM sends a Metafile to PAGES for film or fiche output. PFILM is described in the GR863 writeup.

- Tektronix terminal. PTEKT translates a Metafile sequentially for display on a Tektronix or VT640 terminal. PTEKT is described in the GR865 writeup.

- Metafile editing. PSCAN translates selected pages of a Metafile for display on a Tektronix terminal or a terminal with Tektronix emulation. PSCAN also contains some editing functions that can be used to modify existing Metafiles and to create new Metafiles by copying pages from an existing Metafile. PSCAN is described in the GR864 writeup.

USING CLASSIFIED INFORMATION

General Considerations

If you are not authorized to do classified computing, you can read and write only unclassified files. If you are authorized to do classified computing and you possess a classified password, you can read and write files at any classification up to your maximum classification level.

You must always be conscious of the classification level at which you are signed on to CTSS because files that you create and/or send to an output device will automatically have the same classification level as your signon level. The control key entry <CTRL-E>LL reports your current signon

level if you have forgotten it. (See the Interface document CTSS-1 for a complete listing of control key entries.)

The FILES utility (described in the FILES CTSS-115 document) contains an option that allows you to determine the classification level of your local files. You can then use a utility called SWITCH (described in the SWITCH CTSS-123 writeup) to modify the classification level of your files.

Textual Output

A file of textual information that you generate will be the same classification level as your signon level by default. The PAGES system has been designed to pick up your output file's classification level from the header in the standard print file that is generated by the HSP and FICHE utilities. CBTs handle print output in a similar manner, using the classification level stored in the header generated by the CBT utility.

You have the option to raise the classification of your output file above its original level, but you can not use CBT, FICHE, or HSP to do so. You must use two utilities instead of one for the file conversion and output process: SPRINT to convert native text files to standard print files, and then LCBT, LFICHE, or LHSP to output the print file. The LCBT, LFICHE, and LHSP utilities have a CL option that enables you to raise the classification level of the file. The legal classification levels you can set with CL are as follows.

```
Unclassified (no marking)
OFFICIAL USE ONLY (also Unclassified but marked
                   as OFFICIAL USE ONLY)
SENSITIVE DATA (also Unclassified)
PROTECT AS RESTRICTED DATA
CONFIDENTIAL RD (RD stands for RESTRICTED DATA)
CONFIDENTIAL FRD (FRD stands for FORMERLY RES-
                  TRICTED DATA)
CONFIDENTIAL NSI (NSI stands for NATIONAL SECU-
                  RITY INFORMATION)
SECRET RD (RD same as above)
SECRET FRD (FRD same as above)
SECRET NSI (NSI same as above)
```

Your textual output will be marked automatically with the classification level you have specified. For more information about when and how textual output must be marked with classification levels, see Sections 1 and 2 of the Office Procedures Manual and Section VIII, Chapter 9, of the Los Alamos Security Manual.

Graphics Output

Using the PFILM, PESP, or PCBT postprocessors, you can request that graphics output be marked with an appropriate classification level that is higher than the classification level of the CGS Metafile. (PFILM is described in the GR863 document, PESP in GR862, and PCBT in GR861.)

The allowable classification levels that you can specify with the postprocessors are identical to those that can be set with the LCBT, LFICHE, and LHSP utilities, as described in the previous section. However, please note that at this time, PCBT does not handle sensitive data. The classification level you have chosen will be marked at the top and bottom of each graphics image. For more information about when and how graphics output must be marked with classification levels, see Sections 1 and 2 of the Office Procedures Manual and Section VIII, Chapter 9, of the Los Alamos Security Manual.

LOCATING YOUR PAGES OUTPUT

All PAGES output is placed in the vicinity of the CCF Concourse, which is on the main floor of Building SM-132. Unclassified and classified output are stored in separate locations near the Dispatcher's Desk, which is the focal point of the CCF Concourse.

Unclassified Output

All unclassified output except for OFFICIAL USE ONLY and SENSITIVE DATA output is distributed to shelves located in the CCF Concourse. The OFFICIAL USE ONLY and SENSITIVE output is distributed like PARD output (see the Classified Output section below).

For unclassified output, there are two types of output shelves: user number and organization. Your output will go to your user number shelf by default. The person in your group who is responsible for computing can tell you whether your group and/or division has an organization shelf. If you do have access to an organization shelf, you can phone 7-7927 and request to use the organization shelf as well as your user number shelf. You can then select the organization shelf by entering the <CTRL-E>OD=O control key entry. (Remember control key entries are described in the Interface CTSS-1 writeup.) Many of the output utilities

that we have mentioned in this chapter contain options for overriding the current setting for your output shelf.

User Number Shelf

CCF personnel use the last two digits of your user number to distribute your output to one of 100 shelves, labeled 00-99. For example, if your user number is 69755, your output is placed on shelf 55. Your user number is printed in large numerals on your output to make it easier to locate.

Organization Shelf

An organization label begins with an alphabetic character and appears in large characters on your output. Examples are CDO and WX-11.

Distribution By Types of Output

Film output is on the same shelf (either user number or organization) as the shelf on which your PAGES printer output is placed. Fiche output is stored in an envelope in a metal fiche box on your output shelf. Electrostatic plotter output is stored on the special Versatec output shelves at the west end of the CCF Concourse.

Classified Output

All Secret and Confidential output, both textual and graphics, is stored in a safe in the CCF. The PARD output is stored on shelves directly behind the Dispatcher's desk just inside the CCF. You must ask the Dispatcher to get classified output for you.

USING MAGNETIC TAPE

There are no magnetic tape capabilities on CTSS. If you must use tapes, use the tape facilities on LTSS. Once the tape has been read into your local LTSS file space, you can send the resulting file or files to a CTSS machine by using the MOVE utility. MOVE is described in the MOVE CTSS-119 writeup. LTSS tape-handling is described in the LTSS User's Guide (PIM-6).

5 PRODUCTION

USING THIS CHAPTER

This chapter briefly describes a production system called FOCUS that is available to CTSS users. The PROD utility allows you to communicate with the FOCUS system. For more information about PROD and FOCUS, see the PROD CTSS-22 document.

FOCUS PRODUCTION SYSTEM

FOCUS is a production control system that collects, schedules, controls, and tracks batch production jobs that will be run on the workers. You can run any program that does not require programmer intervention through FOCUS.

This production control system is a combination of several Network components: a central machine called the FOCUS machine, subsystems on the workers, and the Common File System (CFS). The system is reliable because it has duplicate hardware and software components. It also has a manual job scheduling system in case of multiple hardware failures. FOCUS will continue to evolve as new production capabilities are needed.

FOCUS RESOURCES

CTSS machine time for running production jobs is part of your group's or division's allocation of CTSS time. This time is overseen and managed by someone in your group (or division) who is called your queue manager. This person is familiar with the scheduling algorithms used by FOCUS, and can set parameters that determine the conditions under which production jobs in your queue will be run. If you have questions about how you can make better use of production time, you should check with your queue manager.

PROD UTILITY

The PROD utility allows you to submit, retrieve, delete, terminate, abort, and find the status of jobs within FOCUS. Queue managers also use PROD to control the flow of

production jobs within their queues. Through the use of PROD options, you can specify the following parameters.

- CTSS machine and shift on which a job will run
- the maximum amount of CTSS time the job can use
- the maximum amount of memory required
- whether or not the job can be terminated by programmer or operator command
- whether or not the job can be rerun
- the ID of a job that must be completed before the current job can run, if any,
- other pertinent information, including a brief description of a job

COSMOS File

PROD accepts only one form of a production job: a COSMOS input file. COSMOS is an interpretive job control language that can be used to control task executions of system utilities and your programs. With an editor, you can build a COSMOS file consisting of commands to the CTSS utilities that you need. For more information about COSMOS, see the COSMOS CTSS-108 document.

Job Execution

To submit your COSMOS file for execution, use the SUBMIT command in PROD. The production control system will fetch the COSMOS file during the execution of the SUBMIT command and store it on CFS. When the submit process has completed successfully, the production control system will assign a unique job identification and enter it with the job description in the FOCUS machine.

The scheduling of the execution of the job is determined by various algorithms that have been built into FOCUS, but your queue manager can alter the specific scheduling parameters. Once a job has been submitted, you can choose to delete it, modify certain parameters associated with it, abort it if the job has begun execution, force a graceful termination (see the Job Termination section), or retrieve and return it to a particular CFS directory.

Job Monitoring

A PROD command called STATUS gives you an up-to-date status report on a single production job or a class of jobs. The FOCUS machine will contain status information about your job for as long as your job is contained in the FOCUS system and for some time after it has run.

Job Termination

PROD contains two commands for terminating a production job: ABORT and FINISH. ABORT abruptly terminates a job while the job is still running. The ABORT command can also be used to prevent a job from being executed if it has been scheduled to run.

The FINISH command allows you to gracefully terminate an executing job. The graceful termination refers to a condition in which all of your files are closed before being terminated, the files are sent to the device you choose (a PAGES output device, CFS, etc.), and the files are then also usable by subsequent commands in your COSMOS job file. None of these functions, the file closing, file disposing, or file accessibility, will be done when ABORT is used.

Restarting After a System Crash

A production job will be restarted automatically after a system crash. Also, if you specify RR=Y (rerun=yes) when you submit a job, and system problems occur, the CCF operator who oversees the running of production jobs may attempt to rerun your production job from the beginning.

SOME USEFUL RECORD KEEPING TOOLS

There are two CTSS utilities, in addition to the PROD commands, that will help you to keep track of your production jobs and resources. The SNOOPY utility reports bank account usage for production numbers over a period of time that you specify. SNOOPY is described in the SNOOPY CTSS-122 writeup. The QUO utility, which dynamically monitors CTSS system tables and is described in the CTSS-129 writeup, contains a REPOS option that lists CTSS time used for each allocation organization or for selected organizations.

INDEX